# Nelder-Mead and Non-uniform Based Self-organizing Migrating Algorithm

**Dipti Singh and Seema Agrawal**

**Abstract** Self-organizing migrating algorithm (SOMA) is a novel approach capable to solve almost all type of functions. SOMA is highly effective evolutionary optimization technique and has proved its efficiency in solving many real-life applications. This paper presents a new optimization technique M-NM-SOMA to solve global optimization problems. In the proposed algorithm, SOMA is hybridized with Nelder-Mead method as crossover operator and non-uniform mutation operator in order to avoid premature convergence and keep the diversity of the population. The main feature of this algorithm is that it works for very low population size. To authenticate the efficiency of the proposed algorithm, it is tested on 17 benchmark test problems taken from the literature and the obtained results are compared with the results of other existing algorithms. Numerical and graphical results show that M-NM-SOMA has better global search ability and is very efficient, reliable, and accurate in comparison with other algorithms.

**Keywords** Self-organizing migrating algorithm · Nelder-Mead crossover operator · Non-uniform mutation · Particle swarm optimization · Global optimization

## 1 Introduction

A broad class of population-based algorithms for solving global optimization problems has been developed till date. Some of them are genetic algorithms (GA) [1], differential evolution (DE) [2], particle swarm optimization (PSO) [3], ant colony optimization (ACO) [4], and self-organizing migrating algorithm (SOMA)

Dipti Singh
Department of Applied Sciences, Gautam Buddha University,
Greater Noida, India
e-mail: diptipma@rediffmail.com

Seema Agrawal (✉)
Department of Mathematics, S.S.V. College, Hapur, India
e-mail: seemagrwl7@gmail.com

M. Pant et al. (eds.), *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems
and Computing 436, DOI 10.1007/978-981-10-0448-3_66

[5], etc. Among the above mentioned algorithms, SOMA is comparatively a new comer to the class of population-based stochastic search technique capable of handling all type of functions. SOMA can be classified as an evolutionary algorithm, regardless of the fact that no evolution takes place, i.e., no new generations of individuals are created during the search; only the positions of the individuals in the search space are changed during a generation called 'migration loop'. The main features of this algorithm are:

(i) It works efficiently for very low population size.
(ii) It quickly converges to global optimal solutions.

Despite the fact of several attractive features, sometimes SOMA may converge prematurely and the solution may trap to local optima and this situation arises with the increase of dimensionality. As a result, there is diversity loss in the population. To maintain the diversity mechanism, SOMA can be hybridized with local search techniques or other population-based techniques. Hybridization is a grouping of two or more algorithms, in which one seeks a promising region within the large solution space expected to contain global minima, and the other makes use of the search domain to find the best solution rapidly and more precisely. Several attempts have been made earlier to hybridize population-based techniques with other existing approaches [6–12]. First variant of SOMA was developed by Deep and Dipti which is the hybridization of GA and SOMA [13].

Recently, Dipti and Seema developed a number of variants of SOMA, named SOMAQI, SOMA-M, and M-SOMAQI [14–16]. In this paper, a novel variant of SOMA (M-NM-SOMA) based on Nelder-Mead crossover operator and non-uniform mutation operator is proposed. The performance of M-NM-SOMA has been evaluated on the set of 17 benchmark problems and the comparison of it is made with standard PSO and SOMA.

The paper is structured in the following manner: In Sect. 2, preliminaries are presented. M-NM-SOMA is presented in Sect. 3. In Sect. 4, the experimental results are shown. Finally, the paper concludes with Sect. 5 depicting the outcome of the current study.

## 2 Preliminaries

### *2.1 Self-Organizing Migrating Algorithm*

Zelinka and Lampinen [17] first introduced SOMA, which is inspired by the collective behavior of intelligent creatures. This algorithm travels in migration loops and in each migration loop, active individual (individual having worst fitness value) travels a finite distance toward leader (individual having best fitness value) in *N* (path length/step size) moves of defined length (move size). This path is perturbed randomly by perturbation parameter (PRT) which is defined in the range

[0, 1]. Perturbation vector (PRT) controls the perturbation and is created before an individual proceeds toward leader in the following manner:

$$\begin{array}{l}\mathrm{PRTVector}_j = 1 \text{ if } \mathrm{rnd}_j < \mathrm{PRT}, \\ \mathrm{PRTVector}_j = 0, \text{otherwise}\end{array}, \quad j = 1, 2, 3 \ldots n$$

where $\mathrm{rnd}_j$ is uniformly distributed random number in (0, 1) and $n$ is the number of decision variables.

More information regarding SOMA can be obtained from [18].

## 2.2 Nelder-Mead Crossover Operator

The Nelder-Mead simplex search method is a direct search method, originally proposed by Spendley et al. and later modified by Nelder and Mead [19]. First of all, a population is initialized and a simplex is created using ($n$ + 1) points ($n$: the number of variables of a function) chosen arbitrarily from the population. In each migration, the worst point in the simplex is selected first. Then, a new simplex is formed from the old simplex through a sequence of elementary geometric transformations (reflection, contraction, expansion). After each transformation, the current worst point is replaced by a better one. In the proposed algorithm, Nelder-Mead simplex search method is used as a linear Nelder-Mead crossover operator which creates a new point using two out of three randomly chosen points from population.

The computational steps of NM crossover operator method are as follows:

Step1: choose parameters $\gamma > 1$, $\beta > 0$;

Step2: create an initial simplex with randomly chosen three vertices;
find $x_{\mathrm{h}}$ (the worst point), $x_{\mathrm{l}}$ (the best point), $x_{\mathrm{g}}$ (next to the worst point);
calculate their function values $f_{\mathrm{r}}$, $f_{\mathrm{l}}$, $f_{\mathrm{g}}$; the worst point $x_{\mathrm{h}}$ is reflected with respect to the centroid ($x_c$) of other two points;

$$\begin{array}{l} x_{\mathrm{r}} = 2x_c - x_{\mathrm{h.}} \text{ (reflection)} \\ \text{if } f_{\mathrm{r}} < f_{\mathrm{l}} \\ x_{\mathrm{new}} = (1+\gamma)x_{\mathrm{c}} - \gamma\, x_{\mathrm{h.}}\text{(expansion).} \\ \text{else if } f_{\mathrm{r}} \ > = f_{\mathrm{h}} \\ x_{\mathrm{new}} = (1-\beta)x_{\mathrm{c}} + \beta\, x_{\mathrm{h}}. \text{ (contraction).} \\ \text{else if } f_{\mathrm{g}} < f_{\mathrm{r}} < f_{\mathrm{h}} \\ x_{\mathrm{new}} = (1+\beta)x_{\mathrm{c}} - \beta\, x_{\mathrm{h.}} \text{ (contraction).} \end{array} \tag{1}$$

calculate $f_{\mathrm{new}}$ and replace $x_{\mathrm{h}}$ by $x_{\mathrm{new}}$.

Step3: this process continues until termination criterion is satisfied.

### 2.3 Non-uniform Mutation Operator

Non-uniform mutation operator was proposed by Michalewicz [20] to decrease the weakness of random mutation in the real-coded GA. Non-uniform mutation randomly selects one solution $x_{\mathrm{k}}$ from the population and its value is created according to the following rule:

$$\begin{aligned} x'_{\mathrm{k}} &= x_{\mathrm{k}} + (ub_{\mathrm{k}} - x_{\mathrm{k}}) \cdot T \text{ if } \gamma \prec 0.5 \\ \text{Or}\quad x'_{\mathrm{k}} &= x_{\mathrm{k}} + (x_{\mathrm{k}} + lb_{\mathrm{k}}) \cdot T \text{ if } \gamma \geq 0.5 \end{aligned} \tag{2}$$

where $T = \left(\mu\left(1 - \frac{t}{t_{\max}}\right)\right)^{b}$ with $\gamma$ and $\mu$ two uniformly distributed random numbers in the interval [0, 1], $lb_{\mathrm{k}}$ and $ub_{\mathrm{k}}$ are the lower and upper bound of $x_{\mathrm{k}}$, $b > 0$ is a parameter determining the degree of uniformity, $t$ is a migration number, and $t_{\max}$ the maximum number of migrations allowed to run. Non-uniform mutation has fine-tuning capabilities to achieve high precision.

## 3 Proposed Hybrid M-NM-SOMA Algorithm

In this section, a variant of SOMA, M-NM-SOMA has been proposed which is the hybridization of SOMA with Nelder-Mead crossover operator and non-uniform mutation operator. The convergence of standard SOMA is so fast that all other individuals move closer to the best individual very quickly. This causes the population diversity decrease and leads to the premature convergence. To overcome the above problems, SOMA is hybridized with Nelder-Mead crossover operator and non-uniform mutation operator to maintain the diversity among the solutions in the search space.

### 3.1 Methodology of Hybridization

First, the population is initialized randomly spread over the search domain. At each migration the individuals having highest fitness value as leader and having least fitness value as active are selected. Now the active individual travels a finite distance towards leader in N moves of defined length. Among the positions created, the best position is selected and replaces the active individual if it is better than active individual. Now, leader and active individuals are selected again from the population and a new point is created using Nelder-Mead crossover operator using Eq. (1). This new point is accepted only if it is better than active individual and is replaced with active individual. Then leader and active individuals are selected again from the population and a new point is created using non-uniform mutation using Eq. (2). This new point is accepted only if it is better than active individual and is replaced with active individual. The process is continued until some termination criterion is satisfied.

## 4 Experimental Results

The presented algorithm M-NM-SOMA is programmed using C++ and is executed on a Pentium III PC. M-NM-SOMA is used to obtain the results of 17 benchmark problems taken from the literature. All the problems are of minimization with minimum value 0. The seventeen problems with initialization range are given in Table 1. M-NM-SOMA is probabilistic technique and relies a lot on the generation of random numbers; therefore 30 trials of each problem are carried out. A run is measured to be a success if the solution obtained is within 1 % of the preferred precision. The termination criterion of the proposed algorithm in either a run is a success or a preset number of migrations (10,000) are performed.

In order to make a comparative analysis of M-NM-SOMA with SOMA and standard PSO, various performance measures are considered like mean objective function value to check the efficiency and reliability, average number of function evaluations to check the convergence speed, and one more measure success rate is also considered.

The main parameters of M-NM-SOMA are population size, PRT, move size, and path length. The population size is taken as ten for all the problems. PRT parameter varies from 0.1 to 0.9 depending on the problem. The other parameters, move size and path length are taken as 0.31 and 3. Trials for the 17 problems are performed for dimensions (dim) $n$ = 30, 50 and 100.

Table 2 shows successful runs of a total of 30 runs, corresponding to M-NM-SOMA, PSO, and SOMA. Results show that M-NM-SOMA is best in all 17 problems for dim 30 and 100 and it is best in 16 problems for dim 50.

Table 3 shows the average number of function evaluations corresponding to M-NMSOMA, PSO, and SOMA. Results show that M-NM-SOMA is best in 16 problems for all the three dim 30, 50, and 100. Hence on the basis of results, we can say that M-NM-SOMA shows better convergence accuracy.

Table 4 shows the mean objective function value corresponding to M-NM-SOMA, PSO, and SOMA. Results show that M-NM-SOMA is best in 17 problems for dim 30 and is best in 16 problems for dim 50 and 100. Hence, M-NM-SOMA is *most reliable and efficient.* The problems which could not be solved by the particular algorithm is given the symbol (*) at the corresponding entries. The best results are highlighted in bold characters.

Figures 1, 2 and 3 show the mean best objective function value curves for selected benchmark problems and from the figures it is very clear that M-NM-SOMA converges very fast. Hence the presented algorithm M-NM-SOMA shows its superiority over other algorithms PSO and SOMA.

**Table 1** Benchmark functions

| S. no. | Name | Function | Range |
|---|---|---|---|
| 1 | Ackley | $20\exp\left(-0.02\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$ | [−30,30] |
| 2 | Cosine mixture | $0.1n+\sum_{i=1}^{n}x_i^2-0.1\sum_{i=1}^{n}\cos(5\pi x_i)$ | [−1, 1] |
| 3 | Exponential | $1-\left(\exp\left(-0.5\sum_{i=1}^{n}x_i^2\right)\right)$ | [−1, 1] |
| 4 | Griewank | $1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$ | [−600, 600] |
| 5 | Levy and Montalvo-1 | $\frac{\pi}{n}\left(10\sin^2(\pi y_1)+\sum_{i=1}^{n}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_n-1)^2\right), y_i=1+\frac{1}{4}(x_i+1)$ | [−10, 10] |
| 6 | Levy and Montalvo-2 | $0.1\left[\left(\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2\left[1+\sin^2(3\pi x_{i+1})\right]+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right)\right]$ | [−5, 5] |
| 7 | Rastrigin | $10n+\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)\right]$ | [−5.12, 5.12] |
| 8 | Rosenbrock | $\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$ | [−30, 30] |
| 9 | Schewefel-3 | $\sum_{i=1}^{n}\lvert x_i\rvert+\prod\limits_{i=1}^{n}\lvert x_i\rvert$ | [−10, 10] |
| 10 | Step function | $\sum_{i=1}^{n}\left(x_i+{}^{1}\!/_{2}\right)^2$ | [−100, 100] |
| 11 | Dejong’s function with noise | $\sum_{i=1}^{n-1}(i+1)x_i^4+\mathrm{rand}(0,1)$ | [−1.28, 1.28] |
| 12 | Sphere | $\sum_{i=1}^{n}x_i^2$ | [−5.12, 5.12] |
| 13 | Axis parallel hyper ellipsoid | $\sum_{i=1}^{n}ix_i^2$ | [−5.12, 5.12] |
| 14 | Ellipsoidal | $\sum_{i=1}^{n}(x_i-i)^2$ | [−*n*, *n*] |
| 15 | Brown3 | $\sum_{i=1}^{n-1}\left(\left(x_i^2\right)^{\left(x_{i+1}^2+1\right)}+\left(x_{i+1}^2\right)^{\left(x_i^2+1\right)}\right)$ | [−1, 4] |
| 16 | New function | $\sum_{i=1}^{n}\left(0.2x_i^2+0.1x_i^2\sin(2x_i)\right)$ | [−10, 10] |
| 17 | Cigar | $x_1^2+100.00\sum_{i=2}^{n}x_i^2$ | [−10, 10] |

**Table 2** Successful runs of M-NM-SOMA, SOMA and PSO for Dim. 30, 50 and 100

| P. no. | Dimension | No. of successful runs out of 30 | | |
|---|---|---|---|---|
| | | PSO | SOMA | M-NM-SOMA |
| 1 | 30 | 21 | 03 | **30** |
| | 50 | 06 | 0 | **30** |
| | 100 | 0 | 0 | **30** |
| 2 | 30 | 25 | 18 | **30** |
| | 50 | 05 | 0 | **30** |
| | 100 | 0 | 0 | **30** |
| 3 | 30 | 30 | 30 | **30** |
| | 50 | 29 | 30 | **30** |
| | 100 | 0 | 30 | **30** |
| 4 | 30 | 14 | 03 | **29** |
| | 50 | 09 | 07 | **30** |
| | 100 | 0 | 5 | **30** |
| 5 | 30 | 09 | 30 | **30** |
| | 50 | 02 | 28 | **30** |
| | 100 | 0 | 28 | **30** |
| 6 | 30 | 27 | 23 | **30** |
| | 50 | 22 | 19 | **30** |
| | 100 | 0 | 05 | **22** |
| 7 | 30 | 0 | 0 | **10** |
| | 50 | 0 | 0 | **13** |
| | 100 | 0 | 0 | **13** |
| 8 | 30 | 0 | 02 | **25** |
| | 50 | 0 | 0 | **28** |
| | 100 | 0 | 0 | **20** |
| 9 | 30 | 11 | 30 | **30** |
| | 50 | 01 | 30 | **30** |
| | 100 | 0 | 0 | **30** |
| 10 | 30 | 27 | 30 | **30** |
| | 50 | 17 | 30 | **30** |
| | 100 | 0 | 8 | **30** |
| 11 | 30 | 0 | 29 | **30** |
| | 50 | 08 | 30 | **30** |
| | 100 | 0 | 12 | **30** |
| 12 | 30 | 30 | 30 | **30** |
| | 50 | 30 | 30 | **30** |
| | 100 | 01 | 27 | **30** |
| 13 | 30 | 22 | 30 | **30** |
| | 50 | 13 | 30 | **30** |
| | 100 | 0 | 30 | **30** |

(continued)

**Table 2** (continued)

| P. no. | Dimension | No. of successful runs out of 30 | | |
|---|---|---|---|---|
| | | PSO | SOMA | M-NM-SOMA |
| 14 | 30 | 10 | 29 | **30** |
| | 50 | 06 | **29** | 28 |
| | 100 | 03 | 0 | **08** |
| 15 | 30 | 28 | 30 | **30** |
| | 50 | 18 | 30 | **30** |
| | 100 | 0 | 01 | **30** |
| 16 | 30 | 15 | 27 | **30** |
| | 50 | 02 | 26 | **30** |
| | 100 | 0 | 0 | **30** |
| 17 | 30 | 09 | 30 | **30** |
| | 50 | 07 | 30 | **30** |
| | 100 | 0 | 22 | **30** |

**Table 3** Average number of function evaluations of M-NM-SOMA, SOMA, and PSO for Dim. 30, 50, and 100

| P. no. | Dimension | Average no. of function evaluations of successful runs | | |
|---|---|---|---|---|
| | | PSO | SOMA | M-NM-SOMA |
| 1 | 30 | 142,777 | 47,818 | **5861** |
| | 50 | 180,730 | 90,010 | **5824** |
| | 100 | 200,020 | 180,010 | **5490** |
| 2 | 30 | 141,465 | 20,023 | **2897** |
| | 50 | 189,732 | 90,010 | **2992** |
| | 100 | 200,020 | 180,010 | **3391** |
| 3 | 30 | 126,190 | 13,702 | **1963** |
| | 50 | 176,762 | 27,044 | **2068** |
| | 100 | 200,020 | 84,144 | **2383** |
| 4 | 30 | 143,084 | 43,534 | **4168** |
| | 50 | 172,335 | 72,521 | **4195** |
| | 100 | 200,020 | 153,290 | **4198** |
| 5 | 30 | 106,784 | 15,471 | **14,163** |
| | 50 | 143,400 | 27,558 | **26,755** |
| | 100 | 200,020 | 80,829 | **79,537** |
| 6 | 30 | 131,592 | 22,045 | **16,636** |
| | 50 | 165,766 | 46,238 | **23,359** |
| | 100 | 200,020 | 150,842 | **65,635** |

(continued)

**Table 3** (continued)

| P. no. | Dimension | Average no. of function evaluations of successful runs | | |
|---|---|---|---|---|
| | | PSO | SOMA | M-NM-SOMA |
| 7 | 30 | 200,020 | 36,010 | **22,091** |
| | 50 | 200,020 | 90,010 | **10,555** |
| | 100 | 200,020 | 180,010 | **6730** |
| 8 | 30 | 200,020 | 180,010 | **53,014** |
| | 50 | 200,020 | 180,010 | **69,596** |
| | 100 | 200,020 | 180,010 | **97,221** |
| 9 | 30 | 129,625 | 35,712 | **1574** |
| | 50 | 157,040 | 78,148 | **1942** |
| | 100 | 200,020 | 180,010 | **2110** |
| 10 | 30 | 136,301 | 36,372 | **4021** |
| | 50 | 168,018 | 70,822 | **4252** |
| | 100 | 200,020 | 180,010 | **4568** |
| 11 | 30 | 200,020 | 25,285 | **6105** |
| | 50 | 191,830 | 64,604 | **9691** |
| | 100 | 200,020 | 172,555 | **11,851** |
| 12 | 30 | 125,305 | 22,924 | **2278** |
| | 50 | 15,537 | 44,712 | **2467** |
| | 100 | 199,600 | 129,308 | **2931** |
| 13 | 30 | 129,417 | 26,786 | **3886** |
| | 50 | 163,240 | 57,088 | **3391** |
| | 100 | 200,020 | 141,010 | **3622** |
| 14 | 30 | 94,340 | **31,748** | 32,540 |
| | 50 | 125,224 | **70,444** | 79,807 |
| | 100 | 173,520 | **180,010** | 200,010 |
| 15 | 30 | 129,973 | 40,526 | **3055** |
| | 50 | 168,554 | 75,885 | **3055** |
| | 100 | 200,020 | 178,372 | **3323** |
| 16 | 30 | 114,530 | 22,478 | **3580** |
| | 50 | 149,360 | 42,675 | **3350** |
| | 100 | 200,020 | 180,010 | **3875** |
| 17 | 30 | 137,506 | 25,036 | **4945** |
| | 50 | 168,194 | 50,138 | **5115** |
| | 100 | 200,020 | 145,313 | **5428** |

**Table 4** Mean objective function value of M-NM-SOMA, SOMA, and PSO for Dim. 30, 50, and 100

| P. no. | Dimensions | Mean of objective function value of successful runs | | |
|---|---|---|---|---|
| | | PSO | SOMA | M-NM-SOMA |
| 1 | 30 | 0.00975 | 0.00905 | **0.000836** |
| | 50 | 0.00995 | 3.413 | **0.000819** |
| | 100 | 20.016 | 9.06 | **0.000940** |
| 2 | 30 | 0.00937 | 0.00817 | **0.000678** |
| | 50 | 0.00965 | 0.5119 | **0.000722** |
| | 100 | 36.364 | 2.336 | **0.000881** |
| 3 | 30 | 0.00940 | 0.00810 | **0.000556** |
| | 50 | 0.00970 | 0.00940 | **0.000741** |
| | 100 | 0.999 | 0.00810 | **0.000891** |
| 4 | 30 | 0.00960 | 0.00758 | **0.000803** |
| | 50 | 0.00967 | 0.00895 | **0.000855** |
| | 100 | 0.0552 | 0.00940 | **0.000852** |
| 5 | 30 | 0.00892 | 0.00708 | **0.000918** |
| | 50 | 0.00951 | 0.00785 | **0.000944** |
| | 100 | 5.023 | 0.00758 | **0.00658** |
| 6 | 30 | 0.00886 | 0.00884 | **0.000932** |
| | 50 | 0.00912 | 0.00901 | **0.00828** |
| | 100 | 4.327 | 0.00926 | **0.00917** |
| 7 | 30 | 22.866 | 17.32 | **0.00851** |
| | 50 | 76.179 | 35.06 | **0.00765** |
| | 100 | 345.606 | 118.156 | **0.00881** |
| 8 | 30 | 5.38 | 0.0330 | **0.00952** |
| | 50 | 35.175 | 276.092 | **0.00976** |
| | 100 | 195.04 | 1284.4 | **0.00939** |
| 9 | 30 | 0.00967 | 0.00832 | **0.00709** |
| | 50 | 0.00943 | 0.00995 | **0.00774** |
| | 100 | 23.244 | 1.078 | **0.00803** |
| 10 | 30 | 0.00954 | **0.00811** | 0.00859 |
| | 50 | 0.00983 | **0.00882** | 0.00894 |
| | 100 | * | 0.0488 | **0.00860** |
| 11 | 30 | 0.0385 | 0.00703 | **0.00651** |
| | 50 | 0.00981 | 0.00868 | **0.00710** |
| | 100 | * | 0.00969 | **0.00780** |
| 12 | 30 | 0.00992 | 0.00815 | **0.00805** |
| | 50 | 0.00960 | 0.00884 | **0.00881** |
| | 100 | 0.00999 | 0.00771 | **0.00735** |
| 13 | 30 | 0.00950 | 0.00807 | **0.000666** |
| | 50 | 0.00980 | 0.00820 | **0.00789** |
| | 100 | * | 0.00921 | **0.00884** |

(continued)

**Table 4** (continued)

| P. no. | Dimensions | Mean of objective function value of successful runs | | |
|---|---|---|---|---|
| | | PSO | SOMA | M-NM-SOMA |
| 14 | 30 | 0.00956 | 0.00867 | **0.00797** |
| | 50 | 0.00976 | 0.00859 | **0.00815** |
| | 100 | 0.00945 | 6.862 | **0.00565** |
| 15 | 30 | 0.00938 | 0.00814 | **0.000830** |
| | 50 | 0.00978 | 0.00845 | **0.000781** |
| | 100 | * | 0.00984 | **0.000956** |
| 16 | 30 | 0.00964 | 0.00851 | **0.000876** |
| | 50 | 0.00994 | 0.00821 | **0.000907** |
| | 100 | 116.265 | 2.466 | **0.000953** |
| 17 | 30 | 0.00965 | 0.00834 | **0.000766** |
| | 50 | 0.00970 | 0.00885 | **0.000721** |
| | 100 | * | 0.00744 | **0.000889** |

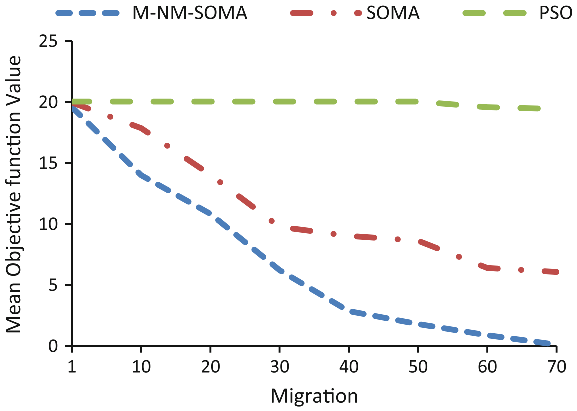


**Fig. 1** Convergence graph of Ackley function for dim. 30

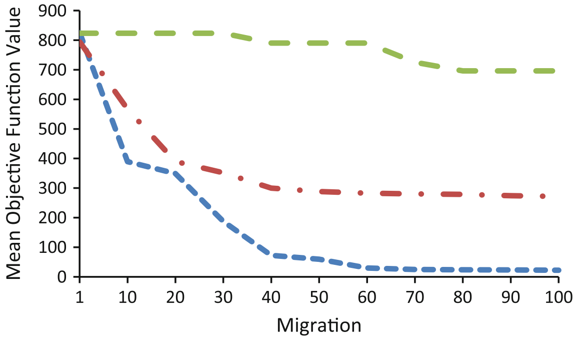
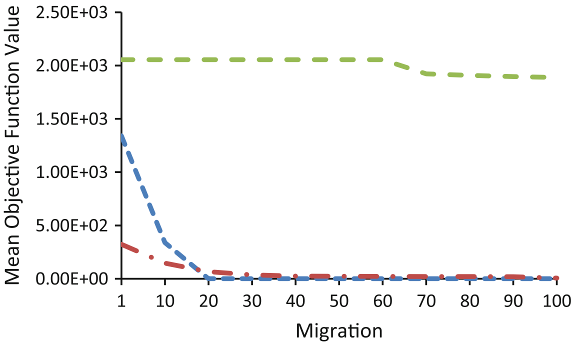


**Fig. 2** Convergence graph of Rastrigin function for dim. 50

**Fig. 3** Convergence graph of Dejong function with noise for dim. 100



## 5 Conclusions

In this paper, a new variant of SOMA, M-NM-SOMA has been proposed. The proposed algorithm is evaluated on 17 unconstrained benchmark problems and obtained results are compared with the results of standard PSO and SOMA. Population size 10 only has been used to evaluate the performance of M-NM-SOMA. On the ground of the results obtained, it can be concluded that the proposed algorithm outperforms PSO and SOMA in terms of population size, efficiency, reliability, accuracy.

## References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesely, Reading (1975)
2. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, IEEE Service Center, Piscataway, pp. 1942–1948 (1995)
3. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 341–359 (1997)
4. Dorigo, M., Stützle, T.: Ant Colony Optimization, MIT Press, 2004
5. Zelinka, I., Lampinen, J.: SOMA- Self organizing migrating algorithm. In: proceedings of the 6th International Mendel Conference on Soft Computing, pp. 177–187, Brno, Czech, Republic (2000)
6. Khosravi, A., Lari, A., Addeh, J.: A new hybrid of evolutionary and conventional optimization algorithm. Appl. Math. Sci. **6**, 815–825 (2012)
7. Pant, M., Thangaraj, R., Abraham, A.: New mutation schemes for differential evolution algorithm and their application to the optimization of directional over-current relay settings. Appl. Math. Comput. **216**, 532–544 (2010) (Elsevier)
8. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. Appl. Math. Comput. **193**, 229–247 (2007)
9. Deep, K., Bansal, J.C.: Hybridization of particle swarm optimization with quadratic approximation. Opsearch **46**, 3–24 (2009) (Springer)

10. Xing, L.N., Chen, Y.W., Yang, K.W.: A novel mutation operator base on immunity operation. European Journal of Operational Research, vol. 197, pp. 830–833 (2009)
11. Deep, K., Das, K.N.: Performance improvement of real coded genetic algorithm with Quadratic approximation based hybridization. Int. J. Intell. Defence Support Syst. **2**, 319–334, (2010) (Inderscience Publisherss)
12. Esmin, A.A.A., Matwin, S.: A hybrid particle swarm optimization algorithm with genetic mutation. Int. J. Innovative Comput. Inf. Control **9**, 1919–1934 (2013)
13. Deep, K., Dipti: A new hybrid self organizing migrating genetic algorithm for function optimization. In: IEEE Congress on Evolutionary Computation, pp. 2796–2803 (2007)
14. Singh, D., Agrawal, S., Singh, N.: A novel variant of self organizing migrating algorithm for function optimization. In: Proceedings of the 3rd international conference on soft computing for problem solving. Advances in intelligent and Soft Computing, vol. 258, pp. 225–234 (2013)
15. Singh, D., Agrawal, S.: Hybridization of self organizing migrating algorithm with mutation for global optimization. In: Proceedings of the international conference on mathematical sciences (ICMS), Elsevier, pp. 605–609 (2014)
16. Singh, D., Agrawal, S.: Hybridization of self organizing migrating algorithm with quadratic approximation and non uniform mutation for function optimization. In: Proceedings of 4th International Conference on Soft Computing for Problem solving, Advances in Intelligent Systems and Computing, Springer, vol. 335, pp. 373–387 (2014)
17. Zelinka, I., Lampinen, J.: SOMA—Self organizing migrating algorithm. In: Proceedings of the 6th International Mendel Conference on Soft Computing, pp. 177–187, Brno, Czech, Republic (2000)
18. Zelinka, I.: SOMA—Self Organizing Migrating Algorithm. In: Onwubolu, G.C., Babu, B.V. (eds.) New optimization techniques in engineering, Springer, Berlin (2004)
19. Nelder, J.A., Mead, R.A.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965)
20. Michalewicz, Z.: Genetic algorithms + Data structures = Evolution programs, 3rd edn. Springer (1996)